HAL
open science

Real-Time Continuous Self-Replicating Details for Shape Deformation

Damien Rohmer, Stefanie Hahmann, Marie-Paule Cani

To cite this version:

Damien Rohmer, Stefanie Hahmann, Marie-Paule Cani. Real-Time Continuous Self-Replicating Details for Shape Deformation. Computers and Graphics, Elsevier, 2015, International Conference Shape Modeling International, 51, pp.67-73. 10.1016/j.cag.2015.05.011 . hal-01152928

HAL Id: hal-01152928

https://hal.inria.fr/hal-01152928

Submitted on 18 May 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Real-Time Continuous Self Replicating Details for Shape Deformation

Damien Rohmer[a,b], Stefanie Hahmann[a], Marie-Paule Cani[a]

[a] *Université Grenoble-Alpes - CNRS (Laboratoire Jean Kuntzmann) - INRIA*
[b] *CPE Lyon, Université de Lyon*

Abstract

We address continuous free-form sculpting of 3D models that carry self-similar geometric details. In addition to being maintained when the model is bent or twisted, repetitive details should be duplicated rather than deformed in stretched regions, so that their distribution and appearance are preserved. Doing so in a temporally coherent way is essential in applications where the user sculpts through continuous deformation gestures. We propose a simple, yet effective solution for achieving such temporal coherence while enabling duplication of details. Our method maintains the set of existing details during stretch but seamlessly grows new ones inbetween. Similarly, some of the details progressively fade out when a region shrinks. Our solution is example-based: it uses the triangles of the initial support surface as exemplars to be selected and re-used in deformed regions. As our results show, our method achieves continuous free-form deformation of complex models while best preserving, at each time step, the properties and appearance of their initial distribution of details.

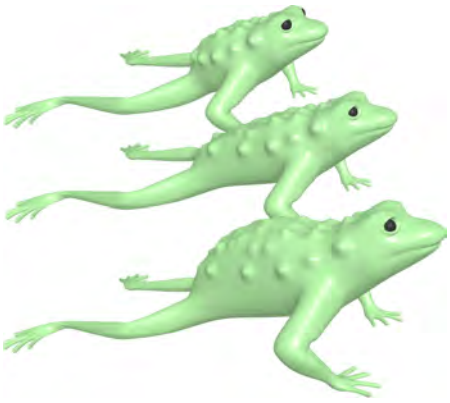

Figure 1: Example of deformation of a surface with continuous duplication of details from the top (original surface) to the bottom.

1. Introduction

Being able to deform 3D models in an intuitive and efficient way is essential for both interactive modeling applications and those related to animation. Many 3D models, especially those representing natural objects are characterized by self-similar details on the surface, as shown in Figure 1. They add an extra level of difficulty when one tries to deform these models, since the details should not to be subject to the same shape deformation to keep the result intuitive. Standard methods for shape deformation manage to preserve the consistence of details as long as the deformation does not include any change in local scale such as stretching or compression, in which cases, the details would also deform and can therefore destroy the natural appearance of the model. For example stretching a tree trunk with bark, a dinosaur wearing protruding scales, or land with trees, will also stretch out those details. In contrast, what the user usually desires is the deformation to be applied only to the base surface, while repetitive details are duplicated in order to maintain the overall appearance of the object. While some approaches for duplicating details have already been proposed, they result into pop-up effects when new details are synthesized.

We present the first method that ensures temporal coherence during mesh deformation while maintaining distibutions of geometric details. This means that during stretching, pre-existing details are maintained and new details are continuously created in between without any *pop-up* effect. Similarly, during compression the surplus details continuously disappear. This allows the user to apply smooth sculpting gestures without being annoyed by visual jumps in the geometry thus improving immersion and the feeling of direct interaction with a shape. We believe that it is one important step towards ”making tools as transparent to the artists as special effects were made transparent to the public’, which is a main remaining challenge in Computer Graphics following R. Cook [1].

Our method tracks the distortion of the surface over time to see when and where new details should be generated or whether existing details should be deleted. In both cases, the actual duplication or deletion process is guided by the extraction of similar exemplars from the original model, enabling us to maintain the expected detail distribution though a transfer process. We handle *natural* objects as opposed to *artificial* (manmade) objects. Our models can be subject to arbitrary 3D continuous deformation except those causing topological changes.

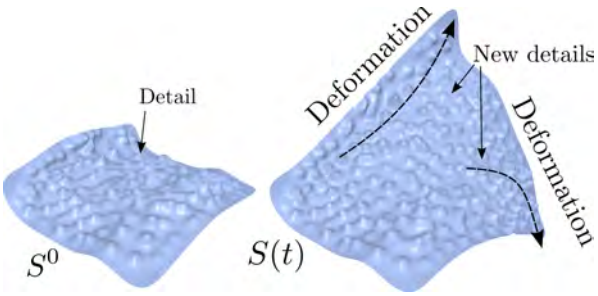

Figure 2: Left: Initial surface S^0 and its details. Right: Surface $S(t)$ after deformation with newly introduced details.

Related Work

Detail preserving 3D deformation. Many mesh deformation methods use an alternative coordinate system to encode the mesh [2, 3, 4]. These approaches have in common linking the orientation of the details to the orientation of the underlying surface so that the details follow the deformation naturally. In case of stretching, the details however undergo the same deformation as the surface, which may lead to unnatural shape distortions. This is particularly true when the amount of stretching is significant.

Kraevoy et al. [5] introduce a non-uniform scaling method able to maintain the important features of a 3D model such as circular parts. Dekkers and Kobbelt [6] avoid undesired detail distortion by extending a successful image resizing technique, called seam-carving [7], to surface meshes. Their "geometry seam-carving" approach preserves the shape and size of salient features and redistributes the distortion over the remaining surface regions between the details. In both methods, the user needs to manually identify the details of interest. Moreover, although great steps towards making deformations intuitive, these methods do not address the problem of preserving distributions of details, which require the introduction of detail duplication/suppression mechanisms, presented next.

Preserving distributions of details. Owada et al. [8] propose a "copy-and-paste" approach to edit a 3D model that carries a distribution of self-similar details. The user defines the zone to be copied and uses guiding curves to set its final shape. This method is however limited to models only having repetitive details along one direction and therefore only allows stretching in that direction. Bokeloh et al. [9] propose a deformation preserving the regular structure of a model. During deformation, the elements of the profile are then continuously inserted or removed. Both works [8, 9] are only applicable to artificially manufactured objects, built of planes, spheres or cylinders of marked symmetry. In our case, we are particularly interested in natural objects whose surfaces are not limited to those sets of basic geometric shapes.

Other methods for preserving distributions of details for 3D shapes build on 2D image editors [10] or texture synthesis methods that handle the preservation of repetitive texture patterns [11]. Bhat et al. [12] generalize texture synthesis to geometric textures, but the method is very costly in computation time. Emilien et al. [13] handles distributions of 3D shapes on a support surface, but with no geometric continuity. Chen et al. [14] and Alhasim et al. [15] introduce a method that non-uniformly resizes a 3D shape by duplicating all the details thanks to a texture synthesis approach. Still, the first one cannot handle non-homogeneous geometrical textures, while the second is limited to one-dimensional deformations described by a skeleton curve embedded into a shape of zero topological genus.

Although they share the same goal of preserving distributions of details during deformation, none of the previous methods tackled the problem of providing temporal coherence during deformation. Our approach is thus a new step towards seamless sculpting of complex shapes.

2. Framework for Self-replicating Details

We introduce a new shape deformation technique able to continuously replicate self-similar non structured details in order to preserve their initial distribution and size along the deformation process (Figure 2). Our method decouples the user-defined deformation of a given base shape from its 3D detail texture and uses an anisotropic detail replication algorithm based on resynthesizing details from the original surface.

Let us denote S^0 the initially given manifold surface mesh, called the *source surface*, but without restriction S^0 can also be be a parametric spline surface or a bivariate height field. Let $f(x,t) \in \mathbb{R}^3$ the deformation

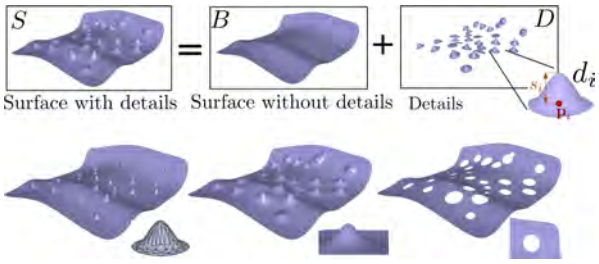

Figure 3: Top: The surface S is represented as a smooth underlying surface B and a distribution of details D. Bottom: Three types of details' geometry: separated mesh (left), heigh field deformer (middle), and holes (right).

process applied to the surface points $x \in \mathbb{R}^3$ at time step $t \geq 0$. We assume the original surface and all deformed surface instances to be represented as a composition of a base surface B and a set of details of interest D, see Figure 3 (top). The initial set of details $D^0 = \{d_i^0\}_{i\in[0,N]}$ is a set of N similar geometric details. Each detail d_i^0 is parameterized by the pair (s_i^0, p_i^0), where s_i^0 is the characteristic size of the detail, and p_i^0 its position in space.

Saying that all details of interest of S^0 are *similar* means that they share the same basic geometry. In our implementation, this basic geometry is a function that procedurally generates the detail either as a 3D mesh added on top of B^0, a procedural function acting as a height field deformer on B^0, or as a hole in the surface B^0, see Figure 3 (bottom). A given set of details parameters D^0 with the associated geometry therefore defines a *distribution* of details over the surface B^0.

The input for our method are the base surface B^0 and the distribution of a set of details of interest D^0. Note that the automatic extraction of the self-similar geometrical details from original model S^0 is beyond the scope of this work. We refer the reader to existing approaches such as symmetries and similaritites analysis [16, 17] and surface decomposition [18].

The deformation is assumed to preserve the surface topology and to be smooth both in space and time to ensure a visually continuous deformation. It could be produced using for instance space deformations, skinning deformations, Laplacian editing or any other procedural or user-driven surface editing technique.

Algorithm overview. Our goal is to determine a sequence of continuously deforming surfaces $S(t)$, so that at any time $t \geq 0$ the distribution of details on the new surface stays approximatively the same as the original distribution D^0 for S^0. This means that when the surface stretches, the existing details do not stretch, but new details smoothly appear instead. We call this process *continuous self-replication of details*.

The general framework to achieve this goal consists of deforming the underlying base surface $B(t)$ following the deformation f at time t, whereas the details distribution and size are updated on top of this. Herein, all positions p_i of details existing at the previous time step $(t-dt)$ are transformed into $p_i(t) = f(p_i(t-dt), t)$, while their size $s_i(t)$ is continuously adapted as well. Moreover, new details can smoothly appear such that the overall distribution is preserved. Our algorithm for time consistent detail generation decomposes into 3 steps:

1. **Analysis** of the deformation from the current distribution of details in order to extract meaningful information about locally decreasing or increasing density of details induced by surface stretch or compression.
2. **Synthesis** of new details, which consists of defining the positions of the new details to be inserted or removed while maintaining the original distribution D^0.
3. Actual **generation** of the current surface $S(t)$ while ensuring **temporal continuity**, i.e. a visually smooth transition of the deformed shape from $(t - dt)$ to t.

3. Analysis of local deformations

The first step in our approach is to measure and quantify the local stretch deformation occurring in the distribution of details using a low resolution triangulation of details' positions. This will determine whether new details need to be inserted or removed.

3.1. Encoding the distribution of details as a mesh

Several surface-based measures can be used to determine whether a surface has been stretched and to quantify the amount of the stretch. Measuring the local variation of surface area is most appropriate for the present application since on the one hand it is an absolute measure and on the other it allows for an efficient implementation of the similarity search algorithm (in Sect. 4.1).

The low resolution triangulation of details $\mathcal{T}^0 = (P^0, C^0)$ is computed by using a Delaunay triangulation of the 2D projections $\tilde{P}^0 = (\tilde{p}_i^0)_{i\in[0,N]}$ of the 3D details positions $P^0 = (p_i^0)_{i\in[0,N]}$. We assume that a parameterization of the set of details of interest P^0 onto a single domain is possible, otherwise, several parameterizations by charts could be used. $\mathcal{T}^0$ can be seen as a coarse triangulation of the surface B^0 with vertices centered on the details. It encodes the relative neighborhood information of each detail and thus enables a characterization of the local properties of the distribution D^0. In particular, if some triangles of $\mathcal{T}^0$ are stretched due to stretch occurring on the base surface $B(t)$ where the details are

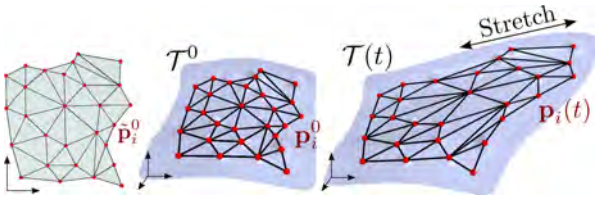

Figure 4: Details' positions in parameterized domain (left) and in 3D space (middle). The details are associated to the triangulation $\mathcal{T}^0$ corresponding to the distribution of details on surface B^0. Right: Deformed triangulation $\mathcal{T}(t)$ after surface stretching. The change of the distribution of the details is encoded in the deformation of $\mathcal{T}(t)$. Its analysis determines if there is need for new details.

anchored, the corresponding region on the surface $S(t)$ is locally losing density of details (see Figure 4) so that the insertion of new details may be required.

Our goal to keep the initial appearance of the distribution of details preserved through time can therefore be redefined more formally as preserving the characteristics of a triangulation of the current vertices $P(t)$, with respect to the original triangulation $\mathcal{T}_0$.

3.2. Computing the distribution deformation

Let $\mathcal{T}(t) = (P(t), C)$ be the current triangulation of details obtained after deformation of the vertex positions $P(t) = (p_i(t))_{i\in[0,N]}$, and C the current mesh connectivity. Both are outputs of the iteration at the previous time step. We define a triangulation $\mathcal{T}(t)$ as elongated when at least one of its triangles has an area larger than every triangle in $\mathcal{T}^0$, i.e. when the following relation holds:

$$\exists\, i\,,\quad \mathcal{A}_i > \alpha\, \mathcal{A}^0_{\max}\,, \tag{1}$$

where $\mathcal{A}_i$ is the area of the ith triangle of $\mathcal{T}(t)$, $\mathcal{A}^0_{\max}$ is the maximal triangle area in $\mathcal{T}^0$, and $\alpha > 1$ is a user defined threshold. Similarly, we define a triangulation $\mathcal{T}(t)$ as compressed when at least one of its triangles has an area smaller than every triangle in $\mathcal{T}^0$

$$\exists\, i\,,\quad \mathcal{A}_i < \beta\, \mathcal{A}^0_{\min}\,, \tag{2}$$

where $\mathcal{A}^0_{\min}$ is the minimal triangle area in $\mathcal{T}^0$, and $0 < \beta < 1$ is another user defined threshold.

When a stretched triangle has been selected using the criteria in (1), it is then a candidate to be subdivided, meaning that at least one new detail will be created inside this triangle. In practice, we precompute $\mathcal{A}^0$, and at run time compute only the areas $\mathcal{A}_i$ of $\mathcal{T}(t)$. Note that in the case where several triangles satisfy condition (1), then the triangle with greatest area is selected to be subdivided. We perform the similar treatment with a compressed triangle selected using criteria (2). In this case, the triangle with smallest area is selected in priority. We define in the next section how we subdivide the selected triangle, or how we remove a vertex from the selected triangle, leading to the synthesis of new geometrical details distribution intended to preserve the original distribution.

4. Synthesis of geometric details

The selected triangle found in the previous section is a stretched or compressed triangle in $\mathcal{T}(t)$. Let us focus first on the stretched case where new details need to be inserted. Remember, that the main goal is to have at each time step a distribution of details similar to the initial one. In order to decide how many details and of which size need to be inserted into the selected stretched triangle $\mathcal{T}(t)$, and in order to compute its, resp. their best positions, we propose a new approach inspired from texture synthesis [19] leading to a self-similar distribution of details' positions.

Let Δ be a stretched triangle of $\mathcal{T}(t)$ according to criteria (1). This triangle has an area larger than every other triangle of $\mathcal{T}^0$ and therefore is associated to a large domain void of details on the surface $S(t)$. The main idea is to replace the triangle Δ by another existing similar triangular domain from the original distribution of details. This large domain may contain itself an interior distribution of details that will then be mapped into the triangle Δ. The triangle Δ will therefore be transformed into a subset of the original distribution, and applying this mapping iteratively will lead to the construction of a general self-similar distribution of details. We explain this algorithm and what we mean with "similar" in the next subsection. Figure 5 illustrates all steps.

4.1. Finding the most similar triangle

First, let us define two triangles Δ and $\tilde{\Delta}$ to be *similar* if they have approximately the same area and the same minimal angle. Note that this criterion is translation and rotation invariant. We then introduce an error of similarity between two triangles Δ and $\tilde{\Delta}$ by

$$E(\Delta, \tilde{\Delta}) = \max\left(|\mathcal{A} - \tilde{\mathcal{A}}|/\mu_{\mathcal{A}}, |\theta - \tilde{\theta}|/\mu_\theta\right), \tag{3}$$

where $\mathcal{A}$, $\tilde{\mathcal{A}}$ denote the area and θ, $\tilde{\theta}$ the minimal angle of Δ and $\tilde{\Delta}$ respectively. $\mu_{\mathcal{A}}$ and μ_θ are respectively normalization factor for the area and the minimal angle.

Let us now call $\mathcal{H}^0$ the set of all triangles formed by the 2D points of $\mathcal{P}^0$ that contain at least one other detail position $\tilde{p}^0_k$ in its interior. Note that by construction the triangles from $\mathcal{T}^0$ do not belong to $\mathcal{H}^0$. Given a

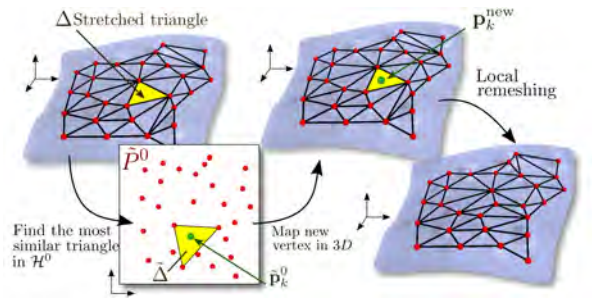

Figure 5: Detail synthesis steps. The stretched 3D triangle in yellow (top left) has a correspondence in the original distribution of details to a similar triangle (bottom left). The relative position of the interior points in green is mapped into the 3D surface (bottom right), and leads to a new distribution of details (top right).

stretched triangle $\Delta \in \mathcal{T}(t)$ we then look for the most similar triangle $\tilde{\Delta} \in \mathcal{H}^0$ by computing

$$\tilde{\Delta} = \text{argmin}_{\tilde{\Delta}_k \in \mathcal{H}^0} \; E(\Delta, \tilde{\Delta}_k)$$

(Figure 5-left). Since the number of triangles of $\mathcal{H}^0$ may be very large ($O(N^3)$), it is critical to perform this step of similarity computation very efficiently to aim for a real time deformation. As pre-processing, we therefore compute $\mathcal{H}^0$ and store each of its triangles in a 2D discrete grid structure whose axes are the triangle area and the minimal angle. Each cell of this grid represents a continuous interval of area and min angle, and stores the list of triangles from $\mathcal{H}^0$ having these characteristics. At run time, finding a list of similar triangles for a given pair $(\mathcal{A}, \theta)$ can therefore be performed very efficiently as a direct query in the discrete grid and on the neighboring discrete cells. Once a conveniently small subset of possible similar candidate triangles is found, the one, minimizing E, is selected by computing explicitly the similarity error E with (3) for each of them.

4.2. Transferring details

Given a stretched triangle $\Delta \in \mathcal{T}(t)$ and its most similar one $\tilde{\Delta}$ in $\mathcal{H}^0$, we can now define the best position to introduce new details. The existence of a triangle $\tilde{\Delta}$ similar to Δ indicates that the same configuration of details exists in the original distribution, and that this configuration contains other details since $\tilde{\Delta}$ contains by construction at least one interior point (see Figure 5).
The general idea consists in reporting the relative coordinates of the interior points of $\tilde{\Delta}$ into the triangle Δ as the position of the newly inserted details, therefore ensuring its local similarity to the original distribution.

Geometrical transfer. Using the barycentric coordinates of each interior point $\tilde{p}_k^0$ with respect to $\tilde{\Delta}$ we get its corresponding 3D position p_k^{new} relative to Δ. To ensure that the new position lies on the underlying base surface, we orthogonally project p_k^{new} onto $B(t)$. Finally, these new vertices are added to the current set P of details positions.

Mesh details connectivity update. The connectivity C of the triangulation $\mathcal{T}$ must also be updated to take into account the insertion of the new details. Still, as no global parameterization of the position of the details is required, we cannot rely on a global incremental Delaunay triangulation. Instead, we locally retriangulate the region around the newly inserted vertices (2-neighborhood of Δ) using only a local parameterization that takes into account the current deformation [20].

4.3. Removing existing details

In the case of compression, we remove an existing vertex of the most compressed triangle Δ from $\mathcal{T}(t)$. In this case, we select the smallest edge of Δ and merge the two vertices into one using an edge collapse operation.

5. Temporal coherence

We have explained so far how to handle the set of positions of the details $P(t)$ so that their distribution stays constantly similar to the original distribution of P^0. This distribution of positions is inherently discontinuous when new details are added or removed. Still, we a looking for a continuous deformation of the surface $S(t)$ even when new details are appearing. When creating new details, our approach relies on setting the size of the newly inserted details to zero so that their topological appearance does not perturb the geometrical shape of the deforming surface. Later we let them grow smoothly as the shape is deformed further. When existing details are removed, we ensure that they first merge continuously to other ones. Moreover, we also aim at a distribution of details' size similar to the original distribution given by $(s_i^0)_{i \in [0,N]}$.
The following subsection describes how we set up the characteristic size $s_i(t)$ of each new detail i during the deformation, how we ensure that details to be removed merge first into other ones, and finally how we display the final resulting surface $S(t)$.

5.1. Continuous detail appearance

We consider that, independently of the choice of the detail's geometrical representation, a detail with zero characteristic size $s_k = 0$ should not visually appear on the geometry of the surface $S(t)$. Therefore, at time t_k,

Figure 6: Temporal coherence during appearance of details.

where the new detail k is inserted, we force $s_k(t_k) = 0$, ensuring that no discontinuity occurs on the resulting surface at time t_k. Next, we make the detail grow smoothly towards a target size s_k^{target}. Considering that the newly inserted detail k corresponds to an original detail k^0 in the distribution D^0, the target size is set to $s_k^{\text{target}} := s_{k^0}^0$. It is thus ensured that all sizes are converging toward an already existing size in the original distribution, and that similar relative positions of details are related to similar sizes of details. Finally, we define a transitional step where the details smoothly grow from zero size to their final target size. Any smooth transition can be used to this end. In our implementation, we use a simple linear interpolation for the transition sizes of the kth detail (see Fig. 6)

$$\forall t \in [t_k, t_k + 1/\mu] \ , \quad s_k(t) = \mu(t - t_k) s_k^{\text{target}} \ ,$$

where μ is a user defined rate of growth for the details.

5.2. Geometrical details merging

Before removing an existing detail, we first ensure that it geometrically merges into another detail. We proceed in the following manner. Let us consider $\mathbf{p}_0(t)$ and $\mathbf{p}_1(t)$ as being the positions of the two extremities of the edge to be merged into a single vertex. Instead of deleting the edge immediately, we apply the following mutual attraction law forcing the two vertices to meet at their middle point $\mathbf{p}_0'(t) = \gamma(\mathbf{p}_1(t) - \mathbf{p}_0(t))$ and $\mathbf{p}_1'(t) = -\mathbf{p}_0'(t)$, where γ is a user defined parameter setting the speed of the attraction displacement. Finally, when the vertices $\mathbf{p}_0(t)$ and $\mathbf{p}_1(t)$ are sufficiently close to each other, we delete one of them, and only perform at this time the local change of the connectivity of the mesh $\mathcal{T}(t)$.

5.3. Geometry of details

To synthesize the final resulting surface $S(t)$, we need the deformed underlying smooth surface $B(t)$, the distribution of details $D(t)$, and the basic geometry that is shared by all the details. We consider in this paper three cases of different geometry for the details: an individual mesh, a height field deformer, and a hole. It is up to the user to implement further geometries as long as they can be generated procedurally using an anchor position and a characteristic size.

6. Results

6.1. Validation

We continuously stretched a surface in a single direction such that its size doubles using three different cases:
Case 1: high density of uniformly distributed details.
Case 2: fewer uniformly distributed details.
Case 3: non uniform distribution with both isolated and grouped details.

Snapshots of results are shown in Figure 7. We considered for all cases $\alpha = 1.1$. We notice that new details are always smoothly appearing throughout the deformation, while previously existing details are consistently following the main deformation of the base surface. It can further be noted that the high density of details is preserved at any time of the deformation in the first case, while in the second case the details remain sparsely distributed on the surface. In the last case, we can notice that new details appear sometimes isolated, and sometimes in a group. In all these cases, the general appearance of the original distribution of details is preserved.

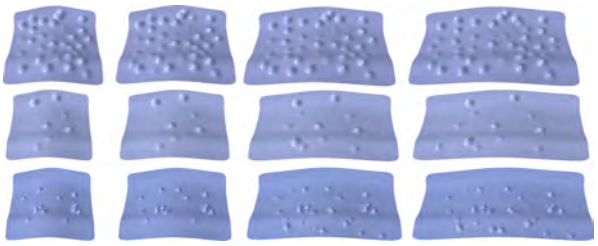

Figure 7: Snapshots of the three scenarios.

We also performed a more quantitative validation of these three cases. Considering the triangulation $\mathcal{T}^0$, we computed the average edge length (avg^0) and the standard deviation of the edge length (std^0) and compared to the maximal and minimal occurring values over all time steps, see Table 6.1. The maximal variation of edge length is smaller than 7% for case 1, and about 15% for cases 2 and 3 with more variability. The variation of the standard deviation of the edge lengths is less than 14% in all cases. Note that the worst case does not necessarily occur at the last time step, but is rather encountered just before a large triangle is subdivided. The error then decreases again immediately after subdivision. These results confirm that our method is able to preserve the average spatial spacing between details.

We also performed a test showing that our method succeeds in preserving simultaneously the distribution of details in the spatial domain and the size of the details, see Figure 8 (left). To this end, we considered a source surface with isolated details which are 3 times

	avg^0	min avg	max avg	std^0	min std	max std
case 1	0.138	0.136	0.147	0.0247	0.0247	0.0271
case 2	0.267	0.263	0.279	0.0681	0.0587	0.0663
case 3	0.225	0.226	0.259	0.181	0.175	0.187

Table 1: Variation of edge-lengths of the coarse triangulation quantifying the distribution of details during the entire deformations of the 3 surfaces in Figure 7.

bigger than the details which are grouped together in clusters. After deforming the surface, we end up with two more isolated details of big size, while three new groups of clustered details are synthesized having small size. Small variability can also be encountered (small isolated details) depending on the most similar triangle found by the algorithm.

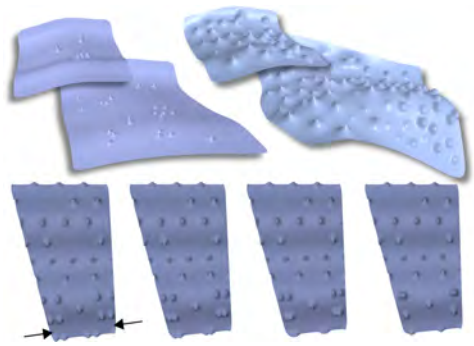

Figure 8: Left: Details organized in clusters (top left). Details of different shapes (top right). Bottom: Compression operation: Neighboring details are continuously merged.

Finally, we also provide an example showing the merging behavior of the details when the surface is locally compressed. As shown in Figure 8 (right) a local compression is applied on the bottom part of the surface, some details associated with small triangles are then continuously merged together.

6.2. Real shapes deformation

We applied our method on more complex shapes with details generated by the three deformers described in Section 5.3. First, a frog shape undergoes some non-homogeneous deformation as shown in Figure 1, where we first stretched the torso of the animal, and then inflated his head-part to round its back. Another example is shown in Figure 9 (top) where we elongated the tail, the neck, and pulled up the back of the dinosaur. The details are procedurally generated by local height-field-bumps computed as a Gaussian function. New holes are appearing and smoothly growing to some individual maximal size. Placed on an existing boat mesh it models a pirate sailboat.

Figure 9: Our method adapts to the deformation of various shapes such as deforming an animal with small scales bumps (top), or modeling efficiently the sail with holes of a pirate ship (bottom).

For all of these examples, an interactive deformation is shown in the associated video including a further whale example using an external mesh to model the details.

6.3. Computation time

Our self-similar detail replication method is very efficient, almost real-time, because it measures local stretch not on the high resolution 3D surface but on a low resolution triangulation formed by the set of details of interest. The analysis of the coarse triangulation $\mathcal{T}^0$ is fast and the comparison of $D(t)$ to D^0 is speeded-up using an acceleration grid structure. We used for all our examples a grid of size 20×20 leading to a very light structure in memory. As an example, for our surface with highest details density of 200 details (Fig.2), the precomputation time (including the generation of the mesh H^0 and the acceleration structure) takes about 500ms for a mesh $\mathcal{H}^0$ of approximately 1.3 millions of triangles. At run time, we separate the computation of the detail structure (computing the new mesh $\mathcal{T}(t)$ at time t when one or several new details are inserted) which is the core of our algorithm, from the actual deformation of geometry explained in Section 5.3. The computation of the detail structure is performed in less than 15ms for up to 200 details (resp. 30ms for up to 500 details).

6.4. Limitations and discussion

Our method succeeds in continuously replicating self-similar details in real-time in response to local surface distortions during surface sculpting.

We cannot handle structured sets of details, such as those aligned along a curve to form a flower or a spiral. The specific case of 1D distribution of aligned details could be handled using the work by Milliez et al. [21], where parts of aligned structured 3D shapes defined by puzzle shape grammars are duplicated.

The deformations we are applying to the shape are not reversible. The fact that we are using non symmetrical operations (birth vs. merge) for compression and stretch created a hysteresis behavior. As for clay, our model behaves as plastic material: the current state depends on the whole history of deformations. If the shape deforms back to its original state, coming back to the same detail positions is not guaranteed. This is however, to our opinion, the price to pay for providing temporal continuity during deformation.

Also, we have demonstrated our concept by implementing three different basic geometries of details. Even though many other procedurally generated detail geometries can be used instead as long as they allow some size control, our method can currently not deal with anisotropic details such as the bark of a tree trunk, wrinkle patterns or hierarchies of details. These extensions are left for future work.

Topology changes are actually excluded, because at the instant where a split of merge takes place, the surface may not be manifold anymore. In practice, however, mesh sculpting systems handling topological changes such as [22] provide an instant change between different manifold parts. So our method could be implemented in this kind of system by re-computing a parameterization of the region(s) of interest just after the topological change.

7. Conclusion and future work

We presented the first method that insures temporal continuity during deformation of complex shapes - including when detail distributions need to be preserved. As our results show, this leads to more immersive interactive sculpting, where the initial distribution of details is seamlessly maintained throughout the deformation.

In future work, we would like to investigate more reversible deformation mechanism, where details appear/disappear in a similar way when the user stretches and then compresses back a given region.

Lastly, our ability to maintain the distribution of details throughout the deformation could be improved by making use of a statistical approach [23, 13] instead of direct extraction from exemplars. This would require extending point processes to enable the preservation of the existing detail footprints while inserting new ones.

Acknowledgements. This work was partially funded by the ERC advanced grant EXPRESSIVE no. 291184 (ERC-2011_ADG_20110209). Many thanks to Niloy Mitra and Michael Wand for interesting discussions and to Nicole Cogo for implementing a preliminary version.

References

[1] R. Cook, Behind the scenes at pixar, SIGGRAPH ASIA, Keynote Talk, 2008.
[2] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, H.-P. Seidel, Laplacian surface editing, in: SGP, 2004, pp. 175–184.
[3] A. Nealen, O. Sorkine, M. Alexa, D. Cohen-Or, A sketch-based interface for detail-preserving mesh editing, ACM TOG 24 (3) (2005) 1142–1147.
[4] Y. Lipman, D. Levin, D. Cohen-Or, Green coordinates, in: ACM TOG (Siggraph), 2008, pp. 78:1–78:10.
[5] V. Kraevoy, A. Sheffer, A. Shamir, D. Cohen-Or, Non-homogeneous resizing of complex models, in: ACM TOG (Siggraph Asia), 2008.
[6] E. Dekkers, L. Kobbelt, Geometry seam carving, CAD 46 (2014) 120–128.
[7] S. Avidan, A. Shamir, Seam carving for content-aware image resizing, ACM TOG (Siggraph) 26 (3) (2007) 1–9.
[8] S. Owada, F. Nielsen, T. Igarashi, Copy-paste synthesis of 3D geometry with repetitive patterns., in: Smart Graphics, Vol. 4073 of LNCS, 2006, pp. 184–193.
[9] M. Bokeloh, M. Wand, V. Koltun, H.-P. Seidel, Pattern-aware shape deformation using sliding dockers, ACM TOG 30 (6) (2011) 123:1–123:10.
[10] H. Fang, J. C. Hart, Detail preserving shape deformation in image editing, in: ACM Siggraph, 2007.
[11] L.-Y. Wei, S. Lefebvre, V. Kwatra, G. Turk, State of the Art in Example-based Texture Synthesis, in: EG STAR Reports, 2009.
[12] P. Bhat, S. Ingram, G. Turk, Geometric texture synthesis by example, in: SGP, 2004, pp. 41–44.
[13] A. Emilien, U. Vimont, C. Marie-Paule, P. Pierre, B. Benes, Worldbrush: Interactive example-based synthesis of procedural virtual worlds, ACM TOG (SIGGRAPH 2015), to appear.
[14] L. Chen, X. Meng, Anisotropic resizing of model with geometric textures, SPM, 2009, pp. 289–294.
[15] I. Alhashim, H. Zhang, L. Liu, Detail-replicating shape stretching, The Vis. Comp. 28 (12) (2012) 1153–1166.
[16] N. J. Mitra, M. Pauly, M. Wand, D. Ceylan, Symmetry in 3D geometry: Extraction and applications, in: EG STAR, 2012.
[17] E. Guy, J.-M. Thiery, T. Boubekeur, Simselect: similarity-based selection for 3D surfaces, CGF (EG) 33 (2) (2014) 165173.
[18] K. Takayama, R. Schmidt, K. Singh, T. Igarashi, T. Boubekeur, O. Sorkine, Geobrush: Interactive mesh geometry cloning, CGF (EG) 30 (2) (2011) 613–622.
[19] P. Bénard, F. Cole, A. Golovinskiy, A. Finkelstein, Self-similar texture for coherent line stylization, NPAR (2010) 1–7.
[20] B. Lévy, S. Petitjean, N. Ray, J. Maillot, Least squares conformal maps for automatic texture atlas generation, ACM TOG 21 (3) (2002) 362–371.
[21] A. Milliez, M. Wand, M.-P. Cani, H.-P. Seidel, Mutable elastic models for sculpting structured shapes, CGF 32 (2013) 21–30.
[22] L. Stãnculescu, R. Chaine, M.-P. Cani, Freestyle: Sculpting meshes with self-adaptive topology, Comput. Graph., (SMI'11) 35 (3) (2011) 614–622.
[23] T. Hurtut, P.-E. Landes, J. Thollot, Y. Gousseau, R. Drouillhet, J.-F. Coeurjolly, Appearance-guided synthesis of element arrangements by example, NPAR (2009) 51–60.